

古文書画像を対象にしたワードスポッティング

寺沢 憲吾[†] 長崎 健[†] 川嶋 稔夫[†]

[†] 公立はこだて未来大学大学院システム情報科学研究科

〒 041-8655 北海道函館市亀田中野町 116-2

E-mail: [†]{g3103004,nagasaki,kawasima}@fun.ac.jp

あらまし 歴史的文書のデジタルアーカイブの構築を考える場合、毛筆手書き文字に対する文書解析手法の開発は必要不可欠である。本研究では毛筆手書き文書画像に対するキーワード検索のための新しい手法として、文字認識手法によらず画像の部分マッチング問題として検索を行う方法を開発し、実験的に有効性を確認する。文字列画像をスリット状に切出すことにより文字列画像はスリット画像のシーケンスとして表現され、さらにこれに固有空間法を適用して低次元化することにより効率的なマッチングが可能となる。また、マッチングに際して DTW (dynamic time warping) を用いることにより、文字の伸縮変形に対応することもできる。

キーワード 毛筆手書き文書, ワードスポッティング, 文字切出し, 固有空間法, DTW

Word Spotting for Historical Document Images

Kengo TERASAWA[†], Takeshi NAGASAKI[†], and Toshio KAWASHIMA[†]

[†] School of Systems Information Science, Future University-Hakodate

Kameda-Nakanochi 116-2, Hakodate-city, Hokkaido, 041-8655 Japan

E-mail: [†]{g3103004,nagasaki,kawasima}@fun.ac.jp

Abstract In creating digital archives of historical documents, it is important to develop effective text retrieval systems for handwritten old characters. This paper describes a new method for text retrieval which requires neither text format transcription nor character segmentation. Instead of segmenting text image into individual characters, the proposed method divides the text image into sequence of small slit style images. By solving matching problem of these sequences, the image region which is corresponding to query image region is retrieved. Applying eigenspace method to the slit images makes it possible to solve the matching problem efficiently. Moreover, using dynamic time warping (DTW) further improves the results.

Key words historical document images, word spotting, segmentation, eigenspace method, DTW

1. はじめに

本研究では毛筆手書き文書画像に対するキーワード検索のための新しい手法として、文字認識手法によらず画像の部分マッチング問題として検索を行う方法を開発し、実験的に有効性を確認する。

地域の図書館や資料館には多くの歴史的資料が貯蔵されており、これらをデジタルアーカイブとして公開し、広く世界に発信して一般の活用を促すことは、学術的文化的観点からのみならず経済的観点からも極めて有益である。実際にデジタルアーカイブを構築することを考える場合、資料をデジタル化して貯蔵する方法だけでなく、貯蔵された情報の中から必要な情報へ素早くアクセスする方法を提供することも主要な技術的課題となるが、歴史的文書のうち特に明治期以前のは毛筆

手書きで書かれたものが多いため従来の文字認識手法の適用が困難であり、自動的な文書解析を行うことができない。そのため現状では特に史料価値の極めて高い文献に対してのみ手作業でインデックス作成が行われているにとどまっている。こうした状況から、歴史的資料におけるデジタルアーカイブの対象をさらに拡大することを考えた場合、歴史的文書画像に対する解析手法の研究開発の必要性は非常に高いものと言える。

一つの方法は、文字認識手法 (OCR) により文書画像をテキスト形式に変換して取り扱うことである。しかし歴史的文書に対して OCR を適用することは極めて困難である。なぜなら毛筆文字は線幅が太く安定した細線化を行いにくいことに加え、崩し字体が多く用いられることや、さらには保存状況による劣化などの問題もあるからである。その結果、OCR の第一段階である文書画像を文字単位に切出すことからして難しいという

のが現状である [1] .

本研究では文字認識ではない別な方法, つまり, 文書画像をテキスト形式に変換することなく画像形式のままで検索を行う手法について検討する. ここでは, 文書画像中からある指定した文字列の部分と類似度の高い部分を検索することを目的とした. これが可能となることにより, 文書画像中から特定のキーワードを含む部分を抽出することができるほか, インデックス作成の作業支援や, あるいは翻刻者が解読できない文字列に遭遇した際にそれと同一の文字列が現れる別の文脈を提示することによる解読支援なども行えると考えている.

1.1 関連研究

本研究で行うような文字認識によらない文字列検索の研究としては, 英語の手書き文書を対象に頻出する単語を抽出した Manmatha ら [2] によるものがあり, “ワードスポッティング”(word spotting) と名づけられている. Rath and Manmatha は, ワードスポッティングに適した 4 つの特徴量を提案し [3], またこれに DTW(dynamic time warping) を適用することにより精度の向上を図っている [4]. また, Marinai ら [5] は主に活字で印刷された古文書を対象に, 自己組織化マップを用いて連結成分を符号化し, 符号列エディット距離を用いて単語間の対応を定める方法を提唱している. これらの研究はいずれも英語の文書を対象としているため, 単語単位の切出しが比較的容易に行われることを前提に, 単語間の対応付けを行っている.

一方で, 日本語や中国語のような言語では単語単位の切出すことが難しい. こうした言語を対象とした研究としては Yue Lu and Chew Lim Tan [6] が中国語の新聞記事を対象とした文字列検索の手法を提案しているものがあるが, これは活字で印刷されたものを対象としており, 文字単位の切出しを行うことが前提となっている.

くずし字やつづけ字などにより文字切出しが困難な文書に対する文字切出しを前提としない研究としては, 探索範囲を単一文字に限定せずに切出すこととした近藤ら [7] の研究がある. ここでは文字幅に着目して探索範囲を切出し, 切出した範囲と正規化した文字パターンとの間でテンプレートマッチングを行うという方法がとられている. また, 著者らは文献 [8] において, 江戸末期から明治期にかけての手書き文書を対象としたワードスポッティングのために, 文字列をスリット状に切出して固有空間法を適用する方法を提唱した.

本研究では, [8] のようにスリット状に文書画像を切出すことを基礎としつつ, これに DTW を適用して文字列の伸縮に対するロバスト性を付加するとともに, 各種パラメータを最適化して精度の向上を図る.

2. スリット切出しによるワードスポッティング

この章では, 文書画像をスリット状に切出すことによって, ある文字列画像からそれに類似した文字列画像を検索する方法の手順について述べる.

2.1 前処理

はじめに, 入力画像に対して前処理を施す. 前処理は,

- 閾値処理により, 背景を消去

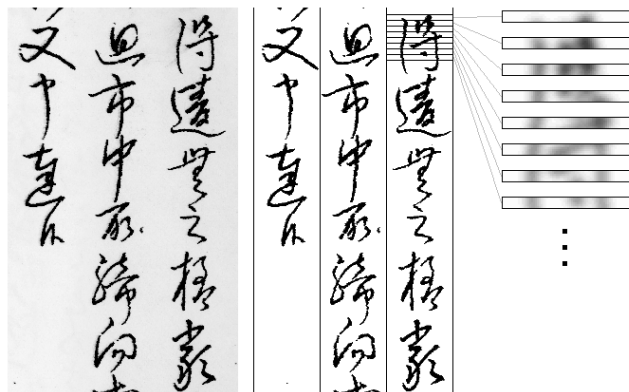


図 1 前処理およびスリット切出し. 左から順に入力画像, 前処理を施したものの, スリット状に切出したもの

- 行の切出し
- 行の中心位置の正規化

の 3 段階で行う.

閾値処理は, 画素値が一定の閾値以下の (すなわち黒に近い) ピクセルのみを有効成分として抽出し, それ以外の部分は背景として画素値を白 (255) にセットするという形で行う. これは二値化処理とは異なって黒い部分の濃淡はそのまま残しており, この後の解析もすべてグレースケールの領域で行う. このような方法を採用したのは, 毛筆画像においてはペン字画像と異なり, 画素値の濃淡に文字識別のために有益な情報が含まれている可能性があるためである.

行の切出しは, 行方向に画素値の射影ヒストグラムを作成することにより行う. 日本語の文書は英語の文書と異なり単語の切出しが容易ではないことはすでに述べたが, しかし行に切出すことは比較的容易である. 行方向の画素値の射影ヒストグラムのピーク位置を行と行との境界と定めることにより, 文書画像を行単位に分割することができる. またここでは同時に, 今後の解析を容易にするため, 左右の余白部分の幅を調節することにより行の幅を一定値にそろえることとする.

最後に, 中心位置の左右への揺れを補正するために, 中心位置の正規化を行う. 罫線の無い紙に書かれた文書は文字位置が左右に揺れる場合がしばしば見られる. この影響を除くため, ある点を中心に行方向に広めの長さのウィンドウを取って文字の重心位置を算出し, 中心点を含んだ行と垂直方向の 1 列について重心が中心に来るように横位置の補正を行う方法により, 中心位置を揃える処理を行う.

これらの処理を行った結果の例を図 1 に示す.

2.2 平滑化およびスリット切出し

前処理済みの画像に対し, 次いでガウシアンフィルタによる平滑化を行った後, これをスリット状に切出す (図 1). ガウシアンフィルタによる平滑化を行うのは, ノイズに対する頑健性を付与するためである. このようにして画像を切出すことにより, 文字列画像をスリット画像のシーケンス (画像列) としてとらえることができるようになる (図 1).

なお, スリット状に切出す際に, 文献 [8] においては切出し開

始位置の影響を小さくするために重なりを含めて切出すこととしているが、本研究ではスリットは重なりを持たせず、単純に一定のピクセル幅毎に切出すこととした。これは、今回の研究にあたってさらに実験を重ねた結果、スリットに重なりを持たせることによる改善効果はほとんど見出せなかったためである。

ここでガウシアンフィルタによる平滑化を行う際のパラメータ σ の値およびスリット状に切出す際の切出し幅については考慮して定める必要があるが、これについては 3.3 節および 3.4 節で検討する。

2.3 特徴量ベクトルの記述

次に、切出した画像列に対し、各スリット画像を低次元の特徴量ベクトルで記述することを考える。本手法では、特徴量ベクトルの記述には、主成分分析（固有空間法）を用いる。

画像における固有空間法の最もよく知られている適用例は顔認識におけるそれであり、Turk and Pentland [9], [10] が顔画像の集合に対して主成分分析を適用して得られた固有顔（Eigenface）を用いることにより効率の良い顔認識が可能であることを示したのをはじめとして、数多くの研究が行われている。ここではまず画像に対する固有空間法の適用法について、簡単にその概要を示す。

M 枚の画像があり、各画像は N 画素をもつものとする。各画像に対し、その画素値を並べて N 次元列ベクトルとして表現したものを x_i とする。各画像から平均画像 $c = (1/M) \sum x_i$ を除去し、それを並べた行列を作成し、

$$A = \begin{pmatrix} x_1 - c & x_2 - c & \cdots & x_M - c \end{pmatrix} \quad (1)$$

とおく。これから共分散行列

$$C = AA^T \quad (2)$$

を作成し、 C に対して固有値問題を解いて、固有値と固有ベクトルを得る。これらを固有値の大きい順に並べ替え、上位の固有ベクトルのみを基底として各画像の低次元表現を得る。すなわち、固有ベクトルを固有値の大きい順に v_1, v_2, \dots として、適当な次元 d までのものを順に並べた行列

$$F = \begin{pmatrix} v_1 & v_2 & \cdots & v_d \end{pmatrix} \quad (3)$$

をつくり、

$$y_i = F^T(x_i - c) \quad (4)$$

とする。

これにより、低次元（ここでは d 次元）の画像の表現 y_i が得られたので対応付けの問題を解くことが容易となる。

実際には、一般に $M \ll N$ であるため AA^T の N 次元固有値問題を直接解くことはせず、代わりに $A^T A$ の M 次元固有値問題に帰着させてから解くという方法が採られる。すなわち、 AA^T の固有ベクトル行列を V 、 $A^T A$ の固有ベクトル行列を U とすると $AU = VD$ が成り立つ（ D は AA^T の固有値の平方根を対角成分に並べた $N \times M$ 行列）ため、 U から V を導くことができる。

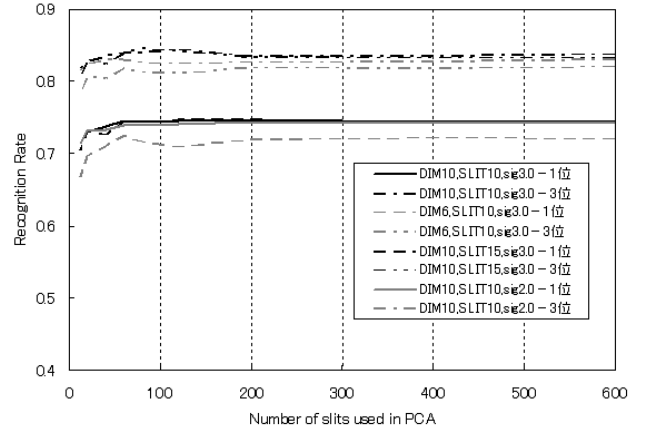


図 2 基底作成に用いるスリット数と、認識率の比較

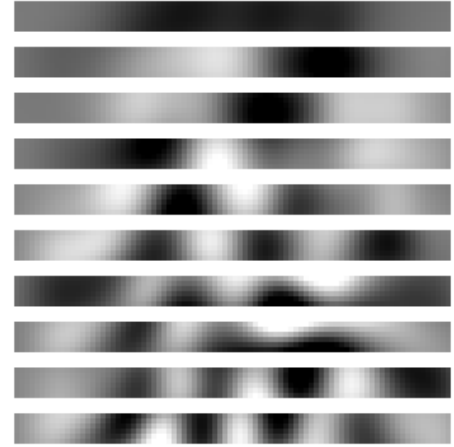


図 3 主成分分析により作成される固有画像の例。実際は正負の値を持つ実数であるが、ここではグレースケールに可視化してある。上から順に第 1 固有画像，…，第 10 固有画像

以上が通常の固有空間法のプロセスであるが、これをそのまま文字列画像から作成したスリット列に適用すると、対象文書の長さ按比例してスリット数（＝画像の枚数・前述のプロセスの M に相当）が増加し、固有値問題の計算が極めて高コストになるという問題が発生する。しかし幸いなことに、今回取り扱うような文字列画像のスリット列はある程度以上スリット数が増えてもほとんど固有画像が変化せず、その結果得られる特徴量ベクトルも変化しないという性質を持っている。それを実験的に確認したのが図 2 であり、この図では基底作成に用いるスリット数を変化させながら、次章で述べるようなさまざまな条件下での認識率の変化の様子を調べたものである。図から、いずれの場合においても認識率は基底作成に用いるスリット数が 50～100 程度の早期に立ち上がり、それ以上スリット数を増やしてもほとんど認識率には影響しないことがわかる。このことから、今後の実験においては基底作成に用いるスリット数は最大 200 スリットとし、これより大きい数のスリットを扱う場合は冒頭の 200 スリットのみから固有空間の基底を作成し、それ以降のスリットについては、こうして作成された基底を用いて特徴量ベクトルに変換することとした。実際に作成される固有画像の例を図 3 に示す。

2.4 特徴量ベクトルの系列による対応付け

前節により、文字列画像を特徴量ベクトルの系列に変換する手法が得られた。この節では、これを用いて文書画像中からクエリ部分と類似度の高い部分を検出する方法について述べる。

スリット画像列の特徴量ベクトルの系列を $\{y(t)\}$ (t はスリット番号) とし、クエリ画像は $t_0 \leq t \leq t_0 + \tau$ の範囲に含まれているものとする。このとき、クエリ画像列 $A = \{y(t)|t_0 \leq t \leq t_0 + \tau\}$ と、 t'_0 を起点とする同じ長さの画像列 $B = \{y(t)|t'_0 \leq t \leq t'_0 + \tau\}$ との間の距離を

$$D(A, B) = \sum_{0 \leq t \leq \tau} |y(t_0 + t) - y(t'_0 + t)| \quad (5)$$

で定め、小さい $D(A, B)$ を与える B をクエリ画像と類似度の高い画像と定義する。ここで $|y(t_0 + t) - y(t'_0 + t)|$ は各スリットにおける特徴量ベクトル間の距離を表し、この定義の方法もいくつかの候補が考えられるが、本研究では最も単純な L1-ノルム (マンハッタン距離)

$$|y(t_0 + t) - y(t'_0 + t)| = \sum_i |y_i(t_0 + t) - y_i(t'_0 + t)| \quad (6)$$

(y_i はベクトル y の第 i 成分を表すものとする) を採用することとした。

B の始点 t'_0 を変化させながら $D(A, B)$ を計算し、最も類似度の高い画像を第 1 位検出画像、以下第 2 位検出画像、第 3 位検出画像、... として出力することとする。これにより、文書画像から、クエリとする部分と類似度の高い部分を検出する方法が得られた。

2.5 Dynamic Time Warping

前節までで文字画像列から類似画像を検出する方法が得られたが、ここではさらにそれを拡張し、文字列の縦方向の伸縮変形に対応するために DTW(dynamic time warping) を導入することを考える。DTW は主に音声認識の分野で発達した手法で、2 つの時系列信号が入力されたときに、それぞれの時間軸を非線形に変形させながら最も良い対応が取れる時間対応を探し、その時間対応の下での類似度を出力するものである。本研究で取り扱う文書画像検索においても、行方向の軸 (縦軸) を時間軸とみなすことによりスリットに分割された文字画像を時系列信号とみなすことができ、DTW を適用することが可能となる (図 4)。以下では DTW の概要と、文字画像に対する適用法について述べる。

時系列信号 $A = \{y(t)|\alpha_1 \leq t \leq \alpha_n\}$ と $B = \{y(t)|\beta_1 \leq t \leq \beta_m\}$ に対し、DTW により時間伸縮を調整した距離 $D(A, B)$ を次のように定義する。

$$D(A, B) = \min \left[\frac{\sum_{\theta=1}^k |y(i_{\theta}) - y(j_{\theta})|}{k} \right], \quad (7)$$

ここで $(i_1, j_1), \dots, (i_k, j_k)$ は対応付けの経路を表し、

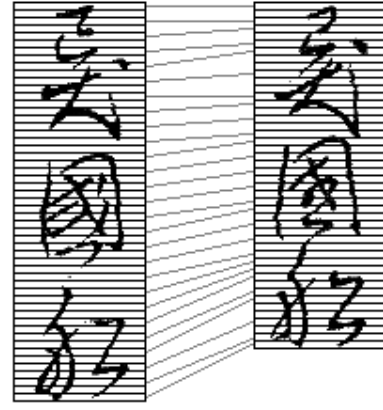


図 4 文字列画像に対する DTW の適用イメージ

$$(i_1, j_1) = (\alpha_1, \beta_1) \quad (8)$$

$$(i_k, j_k) = (\alpha_n, \beta_m)$$

$$(i_{\theta}, j_{\theta}) = \begin{cases} (i_{\theta-1}+1, j_{\theta-1}) \\ (i_{\theta-1}, j_{\theta-1}+1) \\ \text{or } (i_{\theta-1}+1, j_{\theta-1}+1) \end{cases}$$

を満たすものとする。 k は経路長を表す。式 (7) における \min の算出は、あらゆる可能な経路の中で最小のものを求める。可能な経路としては上式を満たす限り無限の伸縮を許容するということでは必ずしもなく、ある一定の範囲に収まるもののみを考える場合が普通である。本研究では、経路は常に次式

$$(1/\alpha) \cdot i_{\theta} \leq j_{\theta} \leq \alpha \cdot i_{\theta}, \quad (9)$$

を満たすものという制約を課した。ここで α は伸縮比を表す。以下の検証では $\alpha = 1.2$ と設定して実験を行う。

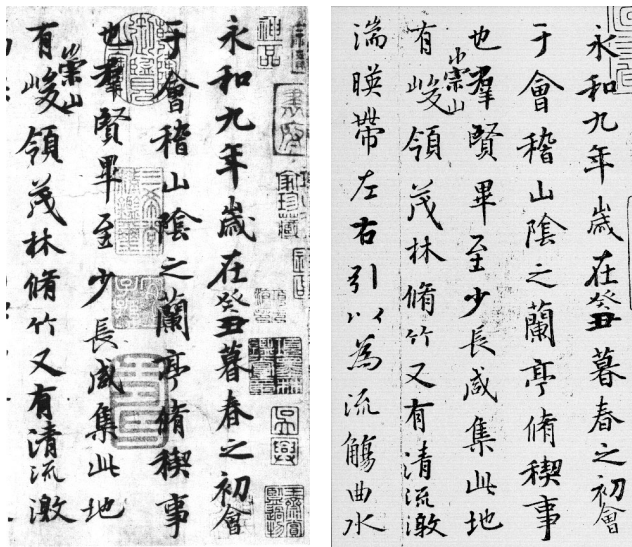
3. 最適パラメータの決定

前章で導入した手順を実際に実装する場合、いくつかのパラメータを決定することが必要となる。この章ではそれらについて検討を行う。

3.1 評価手法

最適なパラメータを適切に検討するためには、システムの性能に関する定量的な指標が必要である。ここではその指標として、王羲之「蘭亭序」の 2 通りの写本 (図 5) に対し、写本 A のある部分をクエリとして写本 B から同一の部分が検出できるか否かを調べ、正しく検出できた割合を認識率として性能評価の指標に用いることとした。「蘭亭序」の 2 通りの写本はいずれも 28 行からなり、321 文字が含まれる。これを 1 文字あたりの解像度をおよそ 200×200 ピクセル程度でスキャンした画像を基礎に、それを人工的にさまざまなレベルに低解像度化したものを対象に、検証を行う。

なお、このサンプルデータに対しては前処理の背景除去の段階で文字列に重なっている印影を画像から除去することができなかったが、これはノイズとしてそのまま残して実験を行っ



(写本 A)

(写本 B)

図 5 評価実験に用いる画像 (部分) . 王羲之「蘭亭序」より . A:神龍半印本, B:張金界奴本

た . したがってこのタスクはやや難しい部類のタスクであると言える .

固有空間および基底の作成にあたっては , 写本 A の冒頭 200 スリットのみを用い , ここで得た基底を全画像に対して適用して特徴量ベクトルを作成した . その後 , 写本 A の任意の連続する 440 ピクセル分の領域 (約 2 文字程度の長さ に相当) をクエリとして写本 B に対して検索を行い , 対応部分が 1 位に検出される割合 (1 位認識率) および 3 位以内に検出される割合 (3 位認識率) を調べた .

なお , この章の評価は各種パラメータの最適値を推定する目的であるため , ここでは DTW を適用せず , 通常の対応付けによる認識率を基準に評価を行った .

3.2 固有空間の次元の決定

まず , 固有空間法で低次元特徴量ベクトルを作成する際の固有空間の次元数を決定するため , 特徴量の次元を変えながら認識率を確認する実験を行った . ここでは文字解像度を 60 ピクセル , スリット数を 10 (すなわちスリット幅 6 ピクセル) とした . その結果を図 6 に示す . 図の横軸は , 特徴量記述に用いた固有空間の次元数を表す . また , 図 7 には固有空間の次元毎の寄与率を示している .

図 6 から , 提案手法による検索は部分空間の次元数が 10 次元程度までの間は次元数につれて増加し , それ以上次元を増やしても認識率の向上には限界があることがわかる . したがってここでは次元 $d = 10$ を採用することとし , これ以降の評価はこれに基づいて行う .

3.3 解像度とスリット幅の決定

次に , 文字列検索における画像の解像度およびスリット切出しの際のスリット幅の影響を確認するため , 解像度とスリット幅を変えながら認識率の変化を調べる実験を行った .

解像度については 1 文字あたり 200 ピクセルの原画像をソフトウェア的に 10 % ~ 50 % に縮小し , 1 文字あたり 20 ~ 100 の

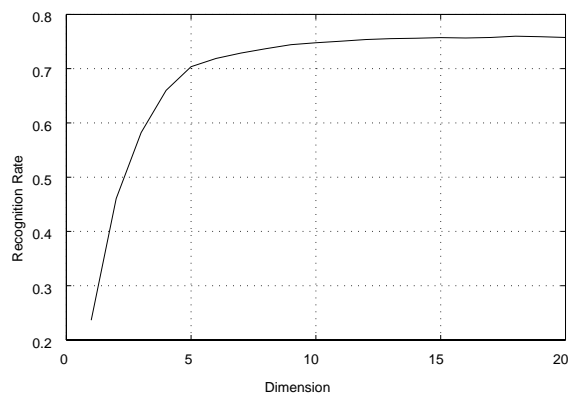


図 6 固有空間の次元と認識率

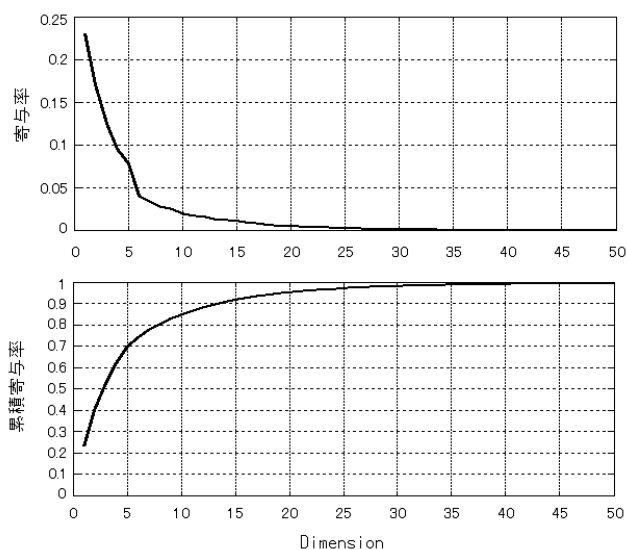


図 7 固有空間の次元毎の寄与率および累積寄与率

解像度の画像を合成した . スリット幅は , 1 文字あたりのスリット数が 4 ~ 20 で , かつスリット幅 2 以上のケースを試した .

結果を図 8 に示す . 等高線図において色の濃い部分が認識率の高い部分である . 図の横軸は 1 文字あたりの解像度 , 縦軸は 1 文字あたりのスリット数を表す . 文字解像度が 60 でスリット数が 10 の場合 , スリット幅が 6 であることを意味している . また , 図 8 には等高線図を解像度 60 で切断した断面と , スリット数 10 で切断した断面も併せて示した .

これらの図から , 解像度は 1 文字あたり 60 ピクセル程度 , スリット数は 1 文字あたり 10 スリット程度取るとほぼ認識率は最大に到達し , それ以上解像度を上げたりスリット幅を細かくしたりしても , 効果は限定的であることがわかる .

3.4 ガウス関数の分散の決定

ここではガウシアンフィルタを適用する際の σ の値に関する実験を行う . 前節の結果を踏まえスリット数は 10 で固定し , 文字解像度を前節同様に変えながら , σ に対する認識率の変化を見た . その結果を示したものが図 9 である .

図から , 解像度が高くなるほど , 最適な σ の値が大きくなる . これはスケールスペースの理論から解釈しても妥当な結果である .

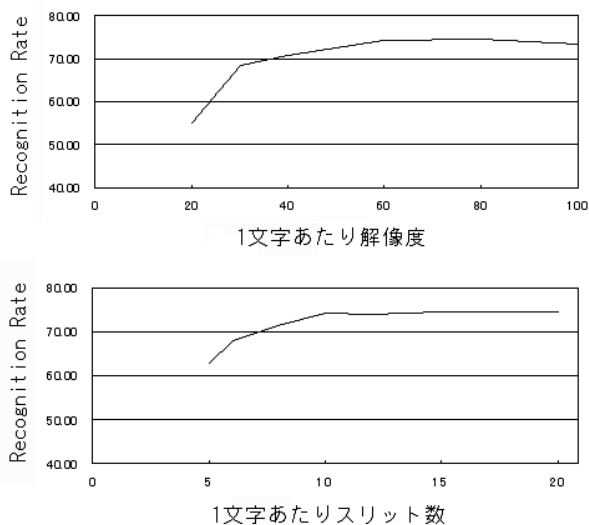
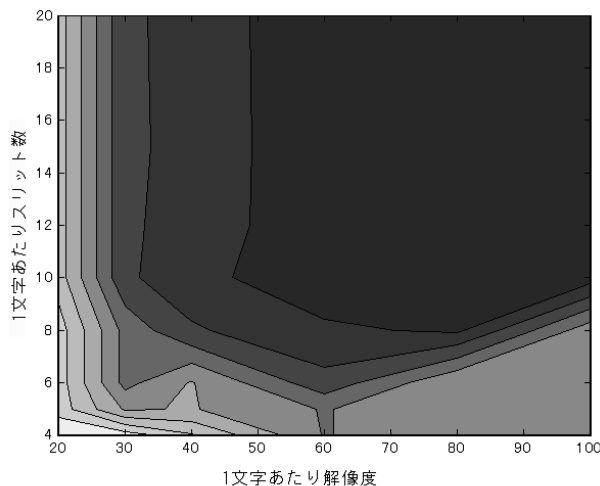


図 8 解像度とスリット数に対する 1 位認識率の等高線図，およびその [スリット数=10] における断面図と [解像度=60] における断面図

具体的には，

$$\sigma = \frac{1 \text{ 文字あたりの解像度}}{20} \quad (10)$$

と設定すると，おおむね最適な結果を得ることができると言える．

3.5 考 察

以上から，1 文字あたりの解像度を 60～80 程度，スリット数は 1 文字あたりで 10 スリット（すなわち幅は 6～8 ピクセル）とし，ガウス関数の値は式 (10) で定めるとよいことがわかる．

1 文字あたり 60 から 80 ピクセル程度というのがどの程度の解像度かを説明するために具体的な例を出すと，これは郵便番号枠（幅 5.7mm）いっぱい文字を 300dpi でスキャンした程度に相当する．実際にデジタルアーカイブの構築を考える際にも，この程度の解像度を得ることは特に問題ない水準であると言えるだろう．

さらに，仮にこれ以下の解像度の画像データしか得られない場合も，単純に画像を拡大して 1 文字あたりのピクセル数を

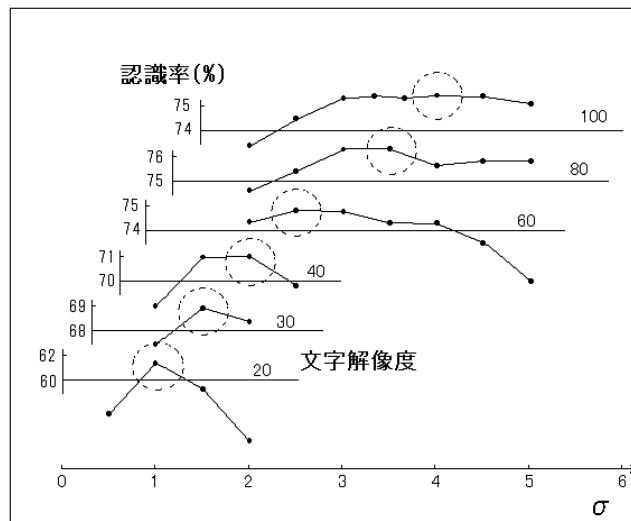


図 9 解像度別のガウス関数の σ の値と 1 位認識率の関係を文字解像度毎に示したもの．图中破線で囲まれた部分は，その解像度における最大認識率を与える部分である．

表 1 低解像度の画像を単純に拡大することによる認識率の改善効果

解像度	1 位認識率 (%)	3 位認識率 (%)
20 (原寸)	61.33	72.55
60 (300%拡大)	71.92	82.75
80 (400%拡大)	72.75	83.73

60～80 程度まで増やしてやることにより，認識率を相当程度押し上げることができる．表 1 はそれを示したもので，解像度 200 の原画像を縮小して解像度を 20 とした画像と，それに拡大処理を施して解像度を 60 または 80 としたものの認識率を比較したものである．ここにおける拡大処理とは一切の補間をせず，300%拡大であれば同じ画素値のピクセルを 3×3 個並べるという極めて単純なものである．表 1 から，このような単純な処理であるにもかかわらず，解像度を 60 または 80 まで拡大することにより，解像度 20 のままで検索を行うのにくらべて認識率が格段に向上することがわかる．

4. 実験結果

4.1 実験 1

前章で得られたパラメタを実際に用いて，改めて王羲之「蘭亭序」の認識率を調べる実験を行った．固有空間は 10 次元，解像度は 1 文字あたり 80（解像度 200 の原画像を 40 %に縮小），スリット数は 1 文字あたり 10（すなわちスリット幅 8 ピクセル），ガウス関数の $\sigma = 4.0$ と設定し，任意の連続する約 2 文字程度の長さの領域（原寸で 440 ピクセル，縮小時で 176 ピクセル）をクエリとして，DTW を適用した場合と適用しない場合の両方について，1 位認識率および 3 位認識率を調べた．その結果が表 2 である．表から，DTW を適用することにより認識率の向上が得られることが確認される．

この実験で得られた 1 位認識率は 78.10%，3 位認識率は 84.43%であるが，ここで認識が正しく行われなかった場合について詳細に調べると，多くの場合，前処理段階において印影の

表 2 DTW を適用した認識率

計算条件	1 位認識率 (%)	3 位認識率 (%)
DTW なし	75.59	84.18
DTW あり	78.10	84.43

表 3 背景をクリアにしたサンプルデータに対する認識率

計算条件	1 位認識率 (%)	3 位認識率 (%)
クエリ長 2-DTW なし	88.70	95.33
クエリ長 2-DTW あり	92.44	96.69
クエリ長 4-DTW なし	93.41	97.50
クエリ長 4-DTW あり	98.38	99.70

影響により背景除去が上手く行われなかった箇所において誤認識が発生していることがわかった．ここで試みに手作業で印影を取り除いて背景をクリアにしたデータを用いて同じ実験を行うと、結果は表 3 のようになる．DTW を適用した場合の認識率は 1 位 92.44%，3 位 96.69%まで向上しており、きわめて高い精度を示していると言える．さらに本手法は文字列と文字列の間の類似度を求めるものであるから、文字列長が長くなるほど認識率が向上するという性質を持っている．ここまでは 2 文字長という比較的短いクエリを与えた場合についての認識率を調べてきたが、クエリ長をこの倍の 4 文字長程度とすれば、認識率は 1 位 98.38%，3 位 99.70%となり、さらに高い精度を示す(表 3 下段)．

4.2 実験 2

これまでの実験は王羲之「蘭亭序」に対して行ってきたが、本手法がこの文献に限って適用可能なものではなく一般性を持つ手法であることを示すために、さらにいくつかの文献を対象に実験を行った結果を示す．

まず、安政元年(1854 年)に書かれた幕府役人の日記である「亜国来使記」(図 10)に対して、キーワード「異国船」を抽出する実験を行った．検索対象画像は 22 ページ、179 行、2771 文字からなる．キーワード「異国船」のうち 1 つをクエリ画像として与え、類似画像を検索した結果が図 11 である．図中、最も左に位置するのがクエリ画像であり、以下順に検出画像が類似度順に並べられている．なおこの図においては白背景の領域のみが実際の検索領域であり、グレー背景の領域は前後の文脈を示すために参考として表示しているものである．この図では 6 位までのうち 5 つが正しい「異国船」である．5 位には「異国船」ではなく「見廻船」が検出されているが、これは最後の文字「船」が一致することにより類似度が高くなったためであると考えられる．

別の例として、比田井小琴「ふみのてほどき」(雄山閣出版)から抜き出した手紙の文書画像(図 12)に対して、二度出現する語句が正確に抽出できるかを調べる実験を行った．この画像は毛筆手書きである上、崩し字・つづけ字が多く見られ、文字切出しを前提とした手法による検索は極めて困難であると思われる．文書は 14 行に 223 文字が含まれ、2 行目と 6 行目に現れる「いたし(い多し)」と、4 行目と 12 行目に現れる「種子」がそれぞれ 2 度出現している．このそれぞれについて、前

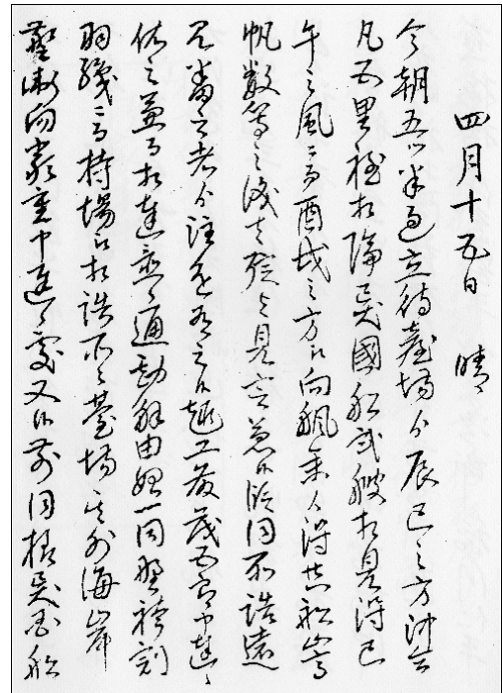


図 10 「亜国来使記」: 安政元年(1854 年)に書かれた幕府役人の日記

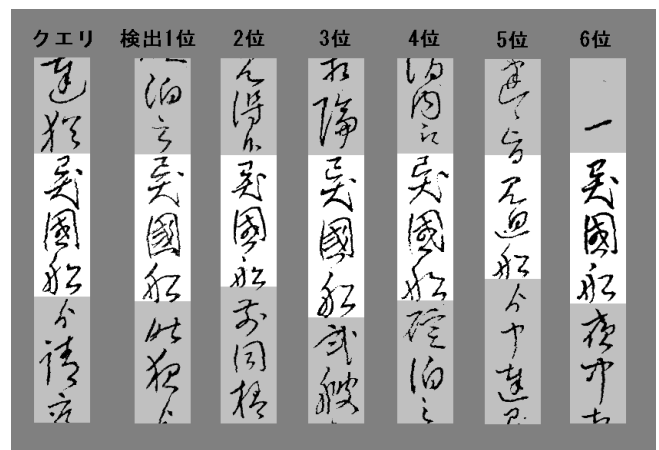


図 11 「亜国来使記」から「異国船」をクエリとして検索を行った結果．白背景の領域が実際の検出部分

者をクエリとして検索を行った．

実験の結果を図 13 に示す．ここでは、最短距離を与える対応画像のほか、次位、第 3 位までのものを示した「いたし」に対する検索結果は、1 位から順に「いたし」「いただき(たく)」「(お出)まし」であり「種子」に対する検索結果は「種子」「(その)折お(話し)」「(皆々)様へも」である．いずれにおいても対応する文字列が第 1 位に正しく検出されていることが確認できる．

5. おわりに

本研究では、毛筆手書きで書かれた文書画像に対し、ある文字列の画像領域をクエリとして与えて、それと類似度の高い画像領域を検索することにより対応する文字列を検出することを目的として、スリット状に切出した上で固有空間法を適用し、

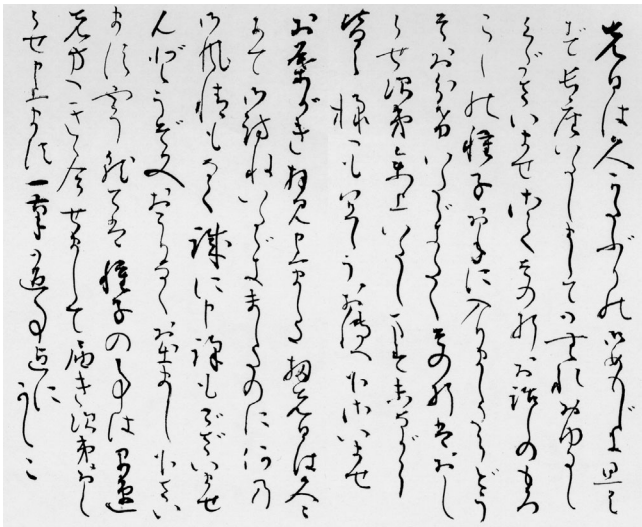


図 12 比田井小琴「ふみのたき」(雄山閣出版)

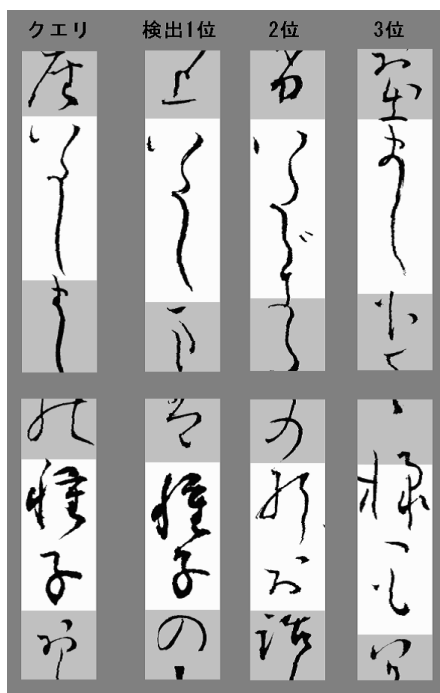


図 13 「ふみのたき」に対する検索結果

シーケンスのマッチング問題として解く方法を提案した。さらに王羲之「蘭亭序」をサンプルデータとして用いて性能の定量的評価を可能とし、各パラメータの最適化を行った。また、文字の文字の行方向の伸縮に対応するために、DTW (dynamic time warping) を導入し、さらに認識率を向上させた。また、文字切出しがきわめて困難な崩し字書体に対しても同手法を適用し、こうした書体に対しても十分に文字列検索が可能であることを示した。

本研究の結果をふまえ、今後は以下のような拡張を予定している。まず第一に、現時点では本手法の適用範囲は同一人物が同一筆跡で文書を書いた場合、あるいはそれに準じる場合にとどまっている。これに対し、筆跡の違いが固有空間および特徴量ベクトルにどのように影響するかを調べ、筆跡の影響を除外

した特徴量ベクトルを作成して異なる筆跡の間で検索を行うことが可能であるかを検証する予定である。

また、本手法により同一文書内の検索が可能となったことにより、何らかの文書構造解析が行える可能性がある。これまでの実験はクエリを手動で与えて類似画像を検出することにとどまっているが、これに対し、あらゆる領域をクエリとしながら検索を行うことにより、文書中から重複語句を抽出できる可能性がある。これは英語のような単語間に区切りが入っている言語においては既往の研究事例があるが、つづけ字・くずし字体で書かれた日本語の毛筆手書き文書に対してはあまり研究事例がない。今後はこうした点に重点を置いて、さらに研究を進めていく予定である。

文 献

- [1] 山田奨治, 柴山守, “古文書を対象にした文字認識の研究,” 情報処理, vol.43, No.9, pp. 950-955, Sep. 2002.
- [2] R. Manmatha, Chengfeng Han and E.M. Riseman, “Word Spotting: A New Approach to Indexing Handwriting,” Proc.of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 631-637, 1996.
- [3] T.M. Rath and R. Manmatha, “Features for Word Spotting in Historical Manuscripts,” Proc.of International Conference on Document Analysis and Recognition, pp. 218-222, 2003.
- [4] T.M. Rath and R. Manmatha, “Word image matching using dynamic time warping,” Proc.of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 521-527, 2003.
- [5] S. Marinai, E. Marino, G. Soda, “Indexing and retrieval of words in old documents”, Proc.of International Conference on Document Analysis and Recognition, pp. 223-227, 2003.
- [6] Yue Lu and Chew Lim Tan, “Word spotting in Chinese document images without layout analysis,” Proc.of IEEE International Conference on Pattern Recognition, pp. 30057-30060, 2002.
- [7] 近藤博人, 松本隆一, 柴山守, 山田奨治, 荒木義彦, “文字切出しを前提としない古文書標題認識,” 情処学研報, no. 2003-CH-57, pp. 1-8, 2003.
- [8] 寺沢憲吾, 長崎健, 川嶋稔夫, “文字切出しによらない毛筆手書き文字検索のための部分空間法,” 信学技報, PRMU2004-172, pp. 51-56, Jan. 2005.
- [9] M.A. Turk and A.P. Pentland, “Eigenfaces for recognition,” Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.
- [10] M.A. Turk and A.P. Pentland, “Face recognition using eigenfaces,” Proc.of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.